DEVICE FOR CONNECTING TWO WORKSTATIONS WITH SEVERAL LINKS

BACKGROUND OF THE INVENTION

1. Technical Field:

The present invention deals with data transfer between two workstations. A workstation is here considered as an equipment of any type provided for exchanging data with another equipment and it could be a personal computer or any kind of terminal.

2. Background Art:

Usually, the data transfer is realized by means of a network, or more precisely with a link of this network. Such a link is generally characterized by the rate at which it forwards data and one distinguishes low speed links and high speed links. In a network, even if several link types are available, the digital link rates are fixed and they often do not fit the optimum rate at which a workstation can transfer data.

When the link rate is lower than this optimum rate, there is a waste of time and the workstation manages the data transfer during a period longer than it should be. When the link rate is higher than this optimum rate, the transmission efficiency of this link is reduced accordingly. Furthermore, the cost of the data transfer is increased since high speed digital links provided by

telecommunication carriers are much more expensive than low speed links.

Document = 09760060

SUMMARY OF THE INVENTION

The present invention therefore concerns a device for connecting a workstation with a network in order to transfer data at a rate close to the optimum rate.

According to the invention, a device is provided for transferring data between two workstations connected to a network; this device comprises means for distributing said data among a plurality of links of said network.

Further, the device comprises a memory for storing said data.

Preferentially, this memory is a dual port static memory.

According to a preferred embodiment, the device comprises:

- a high speed interface for transmitting data from a workstation to the memory,

- associated with each link, a low speed interface for transmitting a part of the data from the memory to the link, and

- a controller for monitoring the data flow between the workstation and the plurality of links, by controlling this memory and these interfaces.

Moreover, the high speed interface receiving data at

an initial rate equal to the sum of the rates at which low speed interfaces transmit on the network, two at least of the low speed interfaces run at different rates.

Preferentially, each low speed interface running at a rate which is a fraction of the initial rate, all these fractions having a common denominator and at least one of these fractions being irreducible, the data flow is cyclically distributed among the low speed interfaces in such a way that each low speed interface receives a number of consecutive bytes from the flow equal to the numerator of its associated fraction.

According to a specific embodiment, at least one of these low speed interfaces comprises means for establishing a connection with a modem.

Likewise, the high speed interface comprises means for transferring data with a modem.

Besides, the device comprises:

- associated with each link, a low speed interface for transmitting part of the data from the link to the memory,

- a high speed interface for transmitting data from the memory to a workstation, and

- a controller for, in a first state, monitoring the data flow between the plurality of links and the workstation by controlling the memory and these

interfaces.

Further, the high speed interface receiving data at an initial rate equal to the sum of the rates at which low speed interfaces receive from the network, two at least of these low speed interfaces run at different rates.

Furthermore, each low speed interface running at a rate which is a fraction of the initial rate, all these fractions having a common denominator and at least one of these fractions being irreducible, the data flow is cyclically distributed among the low speed interfaces in such a way that each low speed interface receives a number of consecutive bytes from the flow equal to the numerator of its associated fraction.

Moreover:

- the high speed interface is provided for alternately transmitting other data from the workstation to the memory,

- each low speed interface is alternately provided for transmitting a part of other data from the memory to the link,

- the controller, in a second state, monitoring the data flow between the workstation and the plurality of links.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the present invention are described below by way of example only with reference to the accompanying drawings, wherein:

Figure 1, the connection of two workstations with a network by means of the device according to the invention,

Figure 2, a diagram of this device,

Figure 3, a diagram of a high speed interface,

Figure 4, a diagram of a dual port static memory,

Figure 5, a diagram of a first low speed interface,

Figure 6, a diagram of a second low speed interface,

Figure 7, a diagram of a third low speed interface,

Figure 8, a diagram of a controller,

Figures 9, **10** and **11** diagrams of logic circuits associated to the controller,

Figure 12, a diagram of a buffer in the static memory,

Figure 13, **14** and **15**, diagrams of translation tables

respectively corresponding to first, second and third programmable read only memory located in the low speed interfaces,

Figure 16, another embodiment of a high speed interface, and

Figure 17, another embodiment of a low speed interface.

Identical elements appearing in several figures are attributed a single reference.

Copy provided by USPTO from the PIRS Image Database on 06/27/2003

DETAILED DESCRIPTION OF THE INVENTION

With reference to **Figure 1**, the invention allows to transfer data between two workstations using three links **11**, **12**, **13** of a network. A first device **D1** is connected from one side with a first workstation A and from the other side with first ends of these three links **11**, **12**, **13**. A second device **D2** is connected from one side with a second workstation **B** and from the other side with the other ends of these links **11**, **12**, **13**. Both devices **D1** and **D2** where lies the present invention are identical.

With reference to **Figure 2**, such a device essentially comprises:

- a memory **5**, a dual port memory according to the preferred embodiment,

- a first **1**, a second **2** and a third **3** low speed interfaces connected from one side with the right port of the memory **5** and from the other side with respectively the first **11**; the second **12** and the third **13** network links,

- a high speed interface **7** connected with a workstation and with the left part of the memory **5**, and

- a controller **8** for controlling the interfaces and the memory.

With reference to **Figure 3**, the high speed interface

7 is described with more details. It is made of a high speed line connector **10**, a line interface unit **20** and a high speed framer **30**. The receive data pin **R** and the transmit data pin **T** of this line connector are respectively connected to the transmit data line and the receive data line of the workstation.

The line interface unit **20** is connected from one side with the receive **R** and transmit **T** data pins of the high speed line connector **10**,and from the other side with the receive **RD** and transmit **TD** data pins of the high speed framer **30**. It converts the signals received from the line connector **10** in TTL signals according to the I.T.U. (International Telecommunication Union) specifications.

The high speed framer **30** takes care of the framing protocol on the high speed link and transmits data received from the line interface unit **20** on a **32** bits high speed data bus **HS_D** with a **24** bits high speed address bus **HS_A**. It is not described in more details since the invention applies to any kind of digital high speed link. One could refer to "TAXI 100 Mbps", "OC3 155 Mbps" or "OC12 622 Mbps" interfaces specified by the I.T.U.

With reference to **Figure 4**, a dual port static memory **40** is connected on its left data port **L_D** and left address port **L_A** respectively with the high speed data bus **HS_D** and address bus **HS_A**. Similarly, the right data port **R_D** and the right address port **R_A** ports are

connected with a 32 bits low speed data bus **PT_D** and with a 24 bits low speed address bus **PT_A**.

With reference to **Figure 5**, the first low speed interface **1** is made of a first low speed line connector **50**, a first line interface device **60** and a first low speed framer **70**.

The first link **11** of the network is connected with the transmit and receive data pins of the first low speed line connector **50**.

The first line interface device **60** is connected from one side with the receive and transmit data pins of the first low speed line connector **50** and from the other side with the receive **RD1** and transmit **TD1** data pins of the first low speed framer **70**. It converts the signals received from the low speed framer in order that they can be transmitted on the network.

The first low speed framer **70** is connected with the low speed data bus **PT_D**. It is also connected with bits 11 to 23 of the low speed address bus **PT_A**. Address bits 0 to 10 are connected to a first programmable read only memory **72** and to a first driver **71**. This driver insulates or transmits the eleven first address bits issued from the first low speed framer **70** according to the state of a signal, a first hold acknowledge signal **HLDA_A** which will be specified later.

With reference to **Figure 6**, similarly, the second

low speed interface **2** is made of a second low speed line connector **80**, a second line interface device **90** and a second low speed framer **100**.

The second link **12** of the network is connected with the transmit and receive data pins of the second low speed line connector **80**.

The second line interface device **90** is connected from one side with the receive and transmit data pins of the second low speed line connector **80** and from the other side with the receive **RD2** and transmit **TD2** data pins of the second low speed framer **100**.

The second low speed framer **100** is connected with the low speed data bus **PT_D**. It is also connected with bits 11 to 23 of the low speed address bus **PT_A**. Address bits 0 to 10 are connected to a second programmable read only memory **102** and to a second driver **101**. This driver is controlled by another signal, a second hold acknowledge signal **HLDA_B** which will be specified later.

With reference to **Figure 7**, similarly, the third low speed interface **3** is made of a third low speed line connector **110**, a third line interface device **120** and a third low speed framer **130**.

The third link **13** of the network is connected with the transmit and receive data pins of the third low speed line connector **110**.

The third line interface device **120** is connected from one side with the receive and transmit data pins of the third low speed line connector **110** and from the other side with the receive **RD3** and transmit **TD3** data pins of the third low speed framer **130**.

The third low speed framer **130** is connected with the low speed data bus **PT_D**. It is also connected with bits 11 to 23 of the low speed address bus **PT_A**. Address bits 0 to 10 are connected to a third programmable read only memory **132** and to a third driver **131**. This driver is controlled by another signal, a third hold acknowledge signal **HLDA_C** which will be specified later.

With reference to **Figure 8**, the controller **8** comprises a microprocessor **140** and several interfaces. The data bus **μP_D** of the microprocessor **140** is connected to the high speed data bus **HS_D** through a first transceiver **141**. This transceiver insulates, transmits from the microprocessor to the bus, or transmits from the bus to the microprocessor according to a driving signal and to a direction signal. The driving signal, a second driving signal **-OE_HS**, and the direction signal, a complementary read signal **-μP_RD** will be detailed later on.

The data bus **μP_D** of the microprocessor **140** is also connected to the low speed data bus **PT_D** through a second transceiver **142**. This transceiver insulates, transmits from the microprocessor to the bus, or transmits from the bus to the microprocessor according to another driving

signal and to the same direction signal. This other driving signal, a first driving signal **-OE_PT** will be detailed later on.

The address bus **μP_A** of this microprocessor is connected to:

- the high speed address bus **HS_A** through a fourth driver **143**, and to.

- the low speed address bus **PT_A** through a fifth driver **144**.

The complementary read **μP_RD** and write **μP_WR** ports of the microprocessor are respectively applied to a sixth **145** and a seventh **146** drivers.

With reference to **Figure 9**, an address decoder **150** is connected to the microprocessor address bus **μP_A**. It further receives a high speed control signal **HS_BUS** and a low speed control signal **PT_BUS** in order to produce a plurality of chip select signals:

- complementary chip select high speed framer - **CS_FMR**, applied to the complementary chip select port of the high speed framer **30**;

- complementary chip select left port of static memory **-CS_LRAM**;

- complementary chip select right port of static

memory **-CS_RRAM;**

- complementary chip select first low speed framer **-CS_FMRA;**

- complementary chip select second low speed framer **-CS_FMRB;**

- complementary chip select third low speed framer **-CS_FMRC;**

- complementary chip select bus select register **-CS_BSR;**

- complementary chip select interrupt controller **-CS_PIC;**

- complementary chip select read only storage - **CS_ROS.**

A first AND gate **151** receives the complementary chip select right port of static memory **-CS_RRAM**, first low speed framer **-CS_FMRA**, second low speed framer **-CS_FMRB** and third low speed framer **-CS_FMRC** for producing the first driving signal **-OE_PT** applied to the second transceiver **142**, the fifth **144** and sixth **145** drivers.

A second AND gate **152** receives the complementary chip select high speed framer **-CS_FMR** and left port of static memory **-CS_LRAM** for producing the second driving

signal **-OE_HS** applied to the first transceiver **141**, the fourth **143** and the seventh **146** drivers.

Besides, a read only storage **ROS 160** is used to store the software. It receives the complementary chip select read only storage **-CS_ROS** and the complementary read signal **μP_RD** produced by the microprocessor **140**. It is also connected to bits 0 to 17 of the microprocessor address bus **μP_A** and to bits 0 to 7 of the microprocessor data bus **μP_D**.

With reference to **Figure 10**, an interrupt controller **170** is connected with bits 0 to 7 of the microprocessor data bus **μP_D** and with bits 0 to 5 of the microprocessor address bus **μP_A**. It produces an interruption signal **INT_μP** applied to the microprocessor **140** from the following signals that is receives:

- the complementary chip select interrupt controller **-CS_PIC**,
- the complementary read signal **-μP_RD**,
- the complementary write signal **-μP_WR**,
- a microprocessor interrupt acknowledge signal INTA from microprocessor **140**,
- a high speed framer interrupt signal **INT_HS** from the high speed framer **30**,
- a first low speed framer interrupt signal **INT_FMRA** from the first low speed framer **70**,
- a second low speed framer interrupt signal **INT_FMRB** from the second low speed framer **100**, and
- a third low speed framer interrupt signal

INT_FMRC from the third low speed framer **130**.

Besides, an arbiter **180** is used to manage the bus requests of the three low speed framers. It receives:

- a first hold signal **HOLD_A** from the first low speed framer **70**,
- a second hold signal **HOLD_B** from the second low speed framer **100**,
- a third hold signal **HOLD_C** from the third low speed framer **130**, and
- a low speed hold acknowledge signal **PT_HLDA** from a device described later on.

It produces :

- the first hold acknowledge signal **HLDA_A** for the first low speed framer **70**,
- the second hold acknowledge signal **HLDA_B** for the second low speed framer **100**,
- the third hold acknowledge signal **HLDA_C** for the third low speed framer **130** and,
- a low speed hold signal **PT_HOLD**.

With reference to **Figure 11**, a register, the bus select register **190** is provided for storing bits 0 to 7 of the microprocessor data bus **μP_D**.

The complementary write input of this register is connected with the output of a first OR gate **153** which receives the complementary write signal **-μP_WR** from the microprocessor **140** and the complementary chip select bus select register **-CS_BSR** from the address decoder **150**.

The complementary read input of the register **190** is connected with the output of a second OR gate **154** which receives the complementary read signal **-μP-RD** from the microprocessor **140** and the complementary chip select bus select register **-CS_BSR** from the address decoder **150**.

A third AND gate **191** receives bits 4 to 7 of the register **190** for producing the low speed control signal **PT_BUS** received by the address decoder **150** (**Figure 9**).

A fourth AND gate **192** receives bits 0 to 3 of the register **190** for producing the high speed control signal **HS_BUS** received by the address decoder.

A fifth AND gate **193** receives the low speed **PT_BUS** and high speed **HS_BUS** control signals for producing a non maskable interruption signal **NMI_μP** received by the microprocessor **140**.

A sixth AND gate **194** receives the high speed control signal **HS_BUS** and a high speed hold signal **HOLD_HS** (from the high speed framer, **30 Figure 3**).

A seventh AND gate **195** receives the low speed control signal **PT_BUS** and the low speed hold signal **PT_HOLD** (from arbiter **180, Figure 10**).

A third OR gate **196** whose inputs are connected to outputs of sixth **194** and seventh **195** AND gates produces a bus request signal **HOLD_μP** which is applied on the hold

input of the microprocessor **140**.

A first multiplexer **197** produces a high speed hold acknowledge signal **HLDA_HS** which is applied on the **HLDA** input of the high speed framer **30** (**Figure 3**). The selection input receives the high speed control signal **HS_BUS**. The first data input receives the high speed hold signal **HOLD_HS** (from the high speed framer **30, Figure 3**). The second data input receives a bus acknowledge signal HLDA_μP generated by the microprocessor 140 on its HLDA output.

A second multiplexer **198** produces the low speed hold acknowledge signal **PT-HLDA** intended for the arbiter **180** (**Figure 10**). The selection input receives the low speed control signal **PT_BUS**. The first data input receives the low speed hold signal **PT_HOLD** from arbiter **180**. The second data input receives the bus acknowledge signal **HLDA_μP**.

The operation of the device according to the invention will now be explained, when a file is transmitted on the network by a workstation.

The previously described architecture allows the microprocessor **140**, the high speed framer **30** and the three low speed framers **70, 100, 130** to work in parallel.

As an example, the address decoder **150** operates according to the following microprocessor address bus **μP_A** states, addresses or data bytes being noted in

hexadecimal:

μP_A comprised between F0000000 and FFFFFFFF: complementary chip select read only storage **-CS_ROS** activated;

μP_A comprised between 70000000 and 7FFFFFFF, and high speed control signal **HS_BUS** equals 1, and low speed control signal **PT_BUS** equals 0, then complementary chip select high speed framer **-CS_FMR** activated;

μP_A comprised between 60000000 and 6FFFFFFF, and high speed control signal **HS_BUS** equals 0, and low speed control signal **PT_BUS** equals 1, then complementary chip select first low speed framer **-CS_FMRA** activated;

μP_A comprised between 50000000 and 5FFFFFFF, and high speed control signal **HS_BUS** equals 0, and low speed control signal **PT_BUS** equals 1, then complementary chip select second low speed framer **-CS_FMRB** activated;

μP_A comprised between 40000000 and 4FFFFFFF, and high speed control signal **HS_BUS** equals 0, and low speed control signal **PT_BUS** equals 1, then complementary chip select third low speed framer **-CS_FMRC** activated;

μP_A comprised between 30000000 and 3FFFFFFF:

complementary chip select register **-CS_BSR** activated;

μP_A comprised between 20000000 and 2FFFFFFF: complementary chip select interrupt controller **-CS_PIC** activated;

μP_A comprised between 10000000 and 1FFFFFFF, and high speed control signal **HS_BUS** equals 1, and low speed control signal **PT_BUS** equals 0, then complementary chip select left port static memory **-CS_LRAM** activated;

μP_A comprised between 00000000 and 0FFFFFFF, and high speed control signal **HS_BUS** equals 1, and low speed control signal **PT_BUS** equals 0, then complementary chip select right port static memory **-CS_RRAM** activated.

The high speed **HS_BUS** and low speed **PT_BUS** control signals are decoded from the bus select register **190**. When this register is programmed with 0F, the fourth AND gate **192** sets the high speed control signal **HS_BUS** to 1. When it is programmed with F0, the third AND gate **191** sets the low speed control signal **PT_BUS** to 1. When the register is programmed by mistake with FF, the fifth AND gate **193** activates the non maskable interruption signal **NMI_μP**.

When this register is cleared, no control signal is activated, which disables the microprocessor **140** to

access high speed or low speed framers.

Therefore, after reset, the read only storage **160** is selected and the microprocessor **140** runs the initialization code. A specific action is to program the internal Direct Memory Access of each framer with buffer transmit and receive addresses. Another action is to clear the bus select register **190**. When the initialization code has been run, the microprocessor enters a wait state until an interruption occurs.

The data flow from a workstation to the network will now be described.

With reference to Figure **12**, workstation A transmits a 2048 bytes file at speed rate F. The first low speed framer **70** operates at speed rate 5F/8, the second one **100** at rate F/4 and the third one **130** at rate F/8.

These bytes are coming from the line connector **10**, they are converted into TTL level by the line interface unit **20** and they are received by the high speed framer **30** in a FIFO (First in, First out) register. When this FIFO register reaches a threshold, the high speed framer **30** requests the bus by activating the high speed hold signal **HOLD_HS**. This hold signal activates the high speed hold acknowledge signal **HLDA_HS** through the first multiplexer **197**, which means that the signal granting the bus to the high speed framer is its own bus request. This allows the microprocessor **140** to control other elements while the high speed framer **30** transfers data to the static memory

40.

When the high speed hold acknowledge signal **HLDA_HS** is activated, the left port of the static memory **40** is selected on its complementary chip select port by means of an inverter **41** and an OR gate **42** intended for realizing the logic operation(-HLDA_HS) + (-CS_LRAM).

Therefore, the high speed framer **30** transfers the data bytes from its FIFO to a receive buffer of the static memory **40**. The base address of this buffer was loaded by the microprocessor **140** during the initialization procedure. When the FIFO is empty, the high speed framer deactivates the high speed hold signal **HOLD_HS**. When the whole file is received and stored in the static memory, the high speed framer **30** activates the high speed framer interrupt signal **INT_HS**. This signal is transmitted to the microprocessor **140** through the interrupt controller **170**, which leads to the execution of the following routine:

- the bus select register **190** is loaded with 0F, which activates the high speed control signal **HS_BUS**, and consequently the bus request signal **HOLD_P**;

- a new receive buffer address is allocated to the high speed framer **30**,

- the three byte counts for the low speed framers are calculated:

. first low speed framer 70 : 2048.5/8 = 1280
. second low speed framer 100 : 2048/4 = 512
. third low speed framer 130 : 2048/8 = 256

- each low speed framer is programmed with its byte count, the base address of the receive buffer (each framer has the same address), and is instructed to start the transmission,

- a framer counter is loaded with **3**, the number of low speed framers.

At the end of this routine, the microprocessor **140** enters again the wait state.

The low speed framers activates their respective hold signals **HOLD_A**, **HOLD_B**, **HOLD_C**, in order to start the transmission. Consequently, the arbiter **180** activates the fourth operate signal **PT_HOLD**, which leads to the activation of the low speed acknowledge signal **PT_HLDA** by the second multiplexer **198**. As seen above, this mechanism allows the microprocessor **140** to control other elements.

When the first low speed framer **70** has the highest priority, the arbiter activates the first hold acknowledge signal **HLDA_A**. This signal activates the static memory **40** on its complementary chip select right port by means of a AND gate which realizes the logic operation (-CS_RRAM).(-HLDA_A).(-HLDA_C). Data bytes can therefore be transferred from the static memory **40** to the first link 11 of the network through the first line

interface device **60**.

This transfer is controlled by the internal Direct Memory Access which generates a complementary read control signal **-RD_PT** and the bytes addresses. The lowest 11 bits of such generated addresses are entered into the first programmable read only memory **72**. This memory **72** which is enabled by the complement of the first hold acknowledge signal **HLDA_A** by means of an inverter **73**, outputs lowest 11 bits of the required address on the low speed address bus **PT_A**. Besides, the highest 13 bits of the required address are directly entered on the low speed address bus. The address translation by means of the programmable memory 72 allows the low speed framer to transmit its own data bytes in the right order.

The translation tables corresponding to first **72**, second **102** and third **132** programmable read only memory are respectively shown in **Figure 13**, **14** and **15**.

When the first low speed framer **70** has loaded its internal transmit FIFO register, it releases the first hold signal **HOLD_A**.

The same transmission process can then be executed by the second **100**, and afterwards by the third **130** low speed framer.

When a low speed framer has completed its transmission, it activates its associated interrupt signal. This signal is forwarded to the microprocessor

140 through the interrupt controller **170** and the following routine is executed:

- the bus select register **190** is loaded with F0 and the low speed hold signal **PT_HOLD** being activated, the bus request signal is therefore activated;

- the microprocessor decrements the framer counter;

- if a new file has been received by the high speed framer **30**, the Direct Memory Access of this low speed framer is reprogrammed to start this new file transmission;

- the bus select register **190** is reset.

At the end of this routine, the microprocessor 140 enters again the wait state. When the framer counter is cleared, it means that the file associated with this counter has been totally transmitted.

It will now be explained how a device operates when a file is received by a workstation from the network.

In an initialization step, the Direct Memory Access of each low speed framer **70, 100, 130** is programmed with the base address of a receive buffer in the static memory **40**. The data bytes come from the three network links **11, 12, 13** at various rates and with different framing protocols, which is managed by the corresponding line interface devices **60, 90, 120** with their associated

framers **70, 100, 130**. Data bytes coming from a link are therefore stored into the FIFO register of the corresponding low speed framer. When this FIFO register reaches a threshold, this framer requests the bus in order to transfer the received data bytes into the static memory **40**. The mode of operation is the same as the one described above dealing with a file transmission from a workstation to the network.

At the end of a reception by a low speed framer, the microprocessor **140** receives an interrupt from this framer through the interrupt controller **170** and the following routine is executed:

- the bus select register **190** is loaded with F0 and, the low speed hold signal **PT_HOLD** being activated, the bus request signal **HOLD_μP** is therefore activated;

- the microprocessor **140** increments a link counter associated with the file reception and stores the number of bytes of this file received by this low speed framer;

- a new receive buffer address is given to the low speed framer, for the next reception;

- the bus select register **190** is reset.

At the end of this routine, the microprocessor **140** enters again the wait state.

The link counter is incremented by each low speed framer at the end of a reception. Therefore, when this counter reaches the number 3, it means that an entire file has been stored into the receive buffer. The following routine is then executed by the microprocessor **140**:

- calculates the number of bytes of this file received by the three low speed framers;

- loads the bus select register **190** with 0F ;

- programs the Direct Memory Access of the high speed framer **30** with the receive buffer address and the total byte count;

- starts the transmission to the workstation;

- resets the bus select register **190**.

The high speed framer **30** therefore transfers data bytes from the static memory **40** to the workstation. The mode of operation previously described remains the same.

At the end of the transmission, the microprocessor **140** receives an interrupt. The main subsequent action is to release the receive buffer which has been used.

According to another embodiment, the invention finds a useful application when the low speed interfaces are connected to the network links **11**, **12**, **13** through low speed modems.

In this case, with reference to **Figure 16**, the high speed interface comprises now a high speed connector **210** instead of the high speed line connector **10**, a workstation interface **220** instead of the line interface unit **20** and a high speed framer **230**.

The high speed connector **210** provides a set of signals that are necessary for the attachment of a modem, essentially a receive data signal **RDM**, a transmit data signal **TDM**, a carrier detect signal **CDM**, a request to send signal **RTS** and a clear to send signal **CTS**.

The workstation interface **220** is connected from one side with the high speed connector **210**, and from the other side with the corresponding ports of the high speed framer **230**. It takes care of the analog characteristics of these signals that, for instance, follow specification V.24 for data rates up to 19.2 Kbps, V.35 for data rates up to 2 Mbps or X.21 for data rates up to 10 Mbps.

The high speed framer **230** is equivalent to this one described with reference to **Figure 3**, just as the high speed connector and the workstation interface are respectively equivalent to the high speed line connector and the line interface.

With reference to **Figure 17**, the first low speed interface now comprises a first low speed connector **250** instead of the first low speed line connector **50**, a first modem interface **260** instead of the first line interface device **60** and a first low speed framer **270**.

The first link **11** of the network is connected with a first modem **280** itself connected with the first low speed connector **250**.

The first modem interface **260** converts the signals received from the low speed framer in order that they can be transmitted on the network.

The first low speed framer **270** takes care of the protocol on the low speed link **11**.

Naturally, the second and third low speed interfaces are modified in the same way as the first one.

The device operates the same way as described in previous modes of operations.

The scope of the present invention is in no way limited to the above embodiments. In particular, any means or steps could be replaced by equivalent means, respectively steps.